

GetAttr

Remember to free memory if numerous calls made

Sean Barnum, Cigital, Inc. [vita¹]

Copyright © 2007 Cigital, Inc.

2007-03-22

Part "Original Cigital Coding Rule in XML"

Mime-type: text/xml, size: 6508 bytes

Attack Category	<ul style="list-style-type: none">• Path spoofing or confusion problem• Denial of Service	
Vulnerability Category	<ul style="list-style-type: none">• Indeterminate File/Path• TOCTOU - Time of Check, Time of Use	
Software Context	<ul style="list-style-type: none">• PLACEHOLDER: AUTHORIZATION	
Location		
Description	<p>The getattr() device configuration subroutine returns the current value of an attribute object or a list of current values of attribute objects from either the Customized Attribute (CuAt) object class or the Predefined Attribute (PdAt) object class. The getattr device configuration subroutine queries the CuAt object class for the attribute object matching the device logical name and the attribute name. It is the application's responsibility to lock the Device Configuration object classes.</p> <p>The getattr subroutine allocates memory for CuAt object class structures that are returned. This memory is automatically freed when the application exits. However, the application must free this memory if it invokes getattr several times and runs for a long time.</p> <p>getattr() references a device by name. As such, any information it reports could be changed by the time the information is acted on, unless the object is locked.</p>	
APIs	FunctionName	Comments
	getattr	
Method of Attack	<p>The key issue with respect to TOCTOU vulnerabilities is that programs make assumptions about atomicity of actions. It is assumed that checking the state or identity of a targeted resource followed by an action on that resource is all one action. In reality, there is a period of time between the check and the use that allows either an attacker to</p>	

1. http://buildsecurityin.us-cert.gov/bsi/about_us/authors/35-BSI.html (Barnum, Sean)

	<p>intentionally or another interleaved process or thread to unintentionally change the state of the targeted resource and yield unexpected and undesired results.</p> <p>An attacker could conceivably change the device referred to by name after getattr() is called and before this information is acted on. This could violate assumptions in the application logic, and consequently open a security hole.</p> <p>If the application does not free memory properly from getattr calls, an attacker could potentially cause repeated getattr calls to the point of resource exhaustion causing a denial of service.</p>		
Exception Criteria			
Solutions	Solution Applicability	Solution Description	Solution Efficacy
	When getattr() is called.	Lock the device prior to calling getattr() and maintain the lock until any actions contingent on the returned information have been performed.	Should be effective.
	Generally applicable.	The most basic advice for TOCTOU vulnerabilities is to not perform a check before the use. This does not resolve the underlying issue of the execution of a function on a resource whose state and identity cannot be assured, but it does help to limit the false sense of security given by the check.	Does not resolve the underlying vulnerability but limits the false sense of security given by the check.
	Generally applicable.	Limit the interleaving	Does not eliminate the

		of operations on files from multiple processes.	underlying vulnerability but can help make it more difficult to exploit.
	Generally applicable.	Limit the spread of time (cycles) between the check and use of a resource.	Does not eliminate the underlying vulnerability but can help make it more difficult to exploit.
	Generally applicable.	Recheck the resource after the use call to verify that the action was taken appropriately.	Effective in some cases.
Signature Details		struct CuAt *getattr (devname, attrname, getall, how_many) char * devname; char * attrname; int getall; int * how_many;	
Examples of Incorrect Code			
Examples of Corrected Code			
Source References		<ul style="list-style-type: none">• ITS4 Source Code Vulnerability Scanning Tool²• http://seclab.cs.ucdavis.edu/projects/vulnerabilities/scriv/ucd-ecs-95-09.pdf³	
Recommended Resources		AIX man page for getattr() ⁴	
DiscriminantSet		Operating System	<ul style="list-style-type: none">• AIX
		Languages	<ul style="list-style-type: none">• C• C++

Cigital, Inc. Copyright

Copyright © Cigital, Inc. 2005-2007. Cigital retains copyrights to this material.

Permission to reproduce this document and to prepare derivative works from this document for internal use is granted, provided the copyright and “No Warranty” statements are included with all reproductions and derivative works.

For information regarding external or commercial use of copyrighted materials owned by Cigital, including information about “Fair Use,” contact Cigital at copyright@cigital.com¹.

The Build Security In (BSI) portal is sponsored by the U.S. Department of Homeland Security (DHS), National Cyber Security Division. The Software Engineering Institute (SEI) develops and operates BSI. DHS funding supports the publishing of all site content.

1. <mailto:copyright@cigital.com>